

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A computer-implemented system to facilitate communication between a client device and a server device comprising:
 - a tabular data stream (TDS) protocol that comprises:
 - a multiple active result set (MARS) header, and
 - a data field that is part of the MARS header and identifies a number of pending requests known by the client device to the server device, the MARS header is employed to transmit, to the server device, the number of pending requests known by the client device to facilitate ~~synchronize~~ synchronization of execution of queries ~~[[for]]~~ to facilitate communication between the client device and the server device, based at least in part on the number of pending requests known by the client device, regardless of buffer size for the client device and the server device.
2. (Original) The system of claim 1, the TDS protocol further comprises a transaction descriptor header that enables a plurality of active transactions under a single session.
3. (Previously presented) The system of claim 1, the TDS protocol further comprises a chunk format component that employs a Partially Length Prefixed (PLP) format to transmit data between the client device and the server device.
4. (Previously presented) The system of claim 1, the TDS protocol further comprises at least one of a transaction state component, reset component, or database

mirroring component, that send information to the client device when a transactional state of the server device changes.

5. (Previously presented) The system of claim 4, the state of the server device changes when a connection is reset to another server device as part of a data base mirror environment.
6. (Previously presented) The system of claim 1, the client device cancels a command being currently executed *via* transmittal of a non severe attention signal without a connection drop of the communication.
7. (Previously presented) The system of claim 1, the client device executes a remote procedure call on the server device.
8. (Previously presented) The system of claim 1, the client device requests a connection to enlist in a distributed transaction coordinator (DTC).
9. (Previously presented) The system of claim 1, the TDS protocol enables a change of order for parameters outputted from the server device, and retrieval of parameters from an application programming interface (API) of the network environment.
10. (Original) The system of claim 1, the TDS protocol specifies a new password as part of a login procedure when an old password is presented.

11. (Currently amended) A computer-implemented system that facilitates communication in client/server networks comprising:
- a server device in communication with a client device *via* a tabular data stream (TDS) protocol in a network environment; and
 - the TDS protocol comprising a query notification header with a data field that requests updates related to a query at a time the communication is initially established to facilitate communication between the server device and the client device, the updates comprise information associated with at least a change to the query.
12. (Currently amended) The system of claim 11 the query notification header establishes channels and setup for the updates sent by the server device to the client device.
13. (Previously presented) The system of claim 11, the query notification header enables at least one of an infrastructure component that can facilitate development of caching layers on top of SQL server applications or the creation of middle tier type caches that remain transparent to the client device.
14. (Withdrawn) A method for establishing communication between a server and a client *via* a tabular data stream (TDS) protocol comprising:
- assigning a major number and a minor number to the TDS protocol based on a release date and a software version thereof;
 - determining the major number and minor number for the client and the server;
 - negotiating down to the TDS protocol common to the server and the client based on comparing respective major numbers and minor numbers; and,
 - initiating handshake *via* the TDS protocol common to the server and the client.
15. (Withdrawn) The method of claim 14, the major number having a 0xNN format and the minor number having a 0x000M format, with M, N being integers.

16. (Withdrawn) The method of claim 14 further comprising providing an increment number as a further designation associated the communication protocol.
17. (Withdrawn) The method of claim 16, the increment number has a 0xFF format, F being a 4-bit hexadecimal value..
18. (Withdrawn) A method for canceling a specific tabular data stream (TDS) request in a client server network comprising:
- sending a non severe attention signal by a client to a server;
 - continuing a read of data by the client;
 - receiving an acknowledgment by the client sent from the server in response to the sending act; and
 - canceling the request without affecting state of a current transaction between the client and the server.
19. (Withdrawn) The method of claim 18 further comprising discarding buffers received by the client after sending the non-severe attention signal.
20. (Withdrawn) The method of claim 19 further comprising discarding buffers received by the client before receiving the acknowledgement.
21. (Withdrawn) A method of changing a password in a client server network comprising:
- specifying a new password as part of a login procedure;
 - verifying a previous password as part of the login procedure; and
 - initiating handshake and session establishment with the new password.

22. (Withdrawn) A method for re-setting a client driver comprising:
setting an attention bit in a TDS header of a packet sent by a server to a client;
notifying the client *via* the attention bit of a desire to abort a current request; and
canceling the current request without dropping an entire connection between the server and the client.
23. (Currently amended) A computer-implemented system that facilitates communication between a client device and a server device comprising:
means for issuing a query by a client device;
means for processing the query by a server device;
means for communicating to the server device a number of pending requests known by the client device; and
means for sending the query results to the client device such that the client device and server device are synchronized, based at least in part on a number of pending requests known by the client device, regardless of a buffer size of the computing system.
24. (Withdrawn) A computing system comprising:
means for issuing a query by a client;
means for processing the query by a server; and
means for sending the query results to the client such that a current command being executed as part of the query is cancelable without dropping a connection established for the query.
25. (Withdrawn) A designator for a tabular data stream (TDS) protocol comprising:
means for identifying a TDS protocol based on a release date and a software version; and
means for determining a TDS protocol common to a server and a client.
26. (Previously presented) The system of claim 1, the TDS protocol further comprising a versioning component that employs a data character scheme to determine a version of a TDS protocol common to the server device and the client device.

27. (Previously presented) The system of claim 26, the versioning component determines a version of a TDS protocol common to the server device and the client device based on chronological versioning and software versioning.

28. (Previously presented) The system of claim 26, the versioning component enables at least one of a down level client device or a new client device to communicate with at least one of a down level server device or a new server device.

29. (Previously presented) The system of claim 3, the chunk format component further employs a special length designator that is communicated to the client device and indicates that a total value of a length of a data stream is not known and the data stream can be transmitted in multiple data stream packets.

30. (Previously presented) The system of claim 4, the database mirroring component enables an application and a client driver to implement a transparent client redirect.